

# Range.java

```
1      /* =====
2      * JFreeChart : a free chart library for the Java(tm) platform
3      * =====
4      *
5      * (C) Copyright 2000-2014, by Object Refinery Limited and Contributors.
6      *
7      * Project Info: http://www.jfree.org/jfreechart/index.html
8      *
9      * This library is free software; you can redistribute it and/or modify it
10     * under the terms of the GNU Lesser General Public License as published by
11     * the Free Software Foundation; either version 2.1 of the License, or
12     * (at your option) any later version.
13     *
14     * This library is distributed in the hope that it will be useful, but
15     * WITHOUT ANY WARRANTY; without even the implied warranty of
16     * MERCHANTABILITY
17     * or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public
18     * License for more details.
19     *
```

19	* You should have received a copy of the GNU Lesser General Public
20	* License along with this library; if not, write to the Free Software
21	* Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301,
22	* USA.
23	*
24	* [Oracle and Java are registered trademarks of Oracle and/or its affiliates.
25	* Other names may be trademarks of their respective owners.]
26	*
27	* -----
28	* Range.java
29	* -----
30	* (C) Copyright 2002-2014, by Object Refinery Limited and Contributors.
31	*
32	* Original Author: David Gilbert (for Object Refinery Limited);
33	* Contributor(s): Chuanhao Chiu;
34	*           Bill Kelemen;
35	*           Nicolas Brodu;
36	*           Sergei Ivanov;
37	*
38	* Changes (from 23-Jun-2001)

39	* -----
40	* 22-Apr-2002 : Version 1, loosely based by code by Bill Kelemen (DG);
41	* 30-Apr-2002 : Added getLength() and getCentralValue() methods. Changed
42	*       argument check in constructor (DG);
43	* 13-Jun-2002 : Added contains(double) method (DG);
44	* 22-Aug-2002 : Added fix to combine method where both ranges are null, thanks
45	*       to Chuanhao Chiu for reporting and fixing this (DG);
46	* 07-Oct-2002 : Fixed errors reported by Checkstyle (DG);
47	* 26-Mar-2003 : Implemented Serializable (DG);
48	* 14-Aug-2003 : Added equals() method (DG);
49	* 27-Aug-2003 : Added toString() method (BK);
50	* 11-Sep-2003 : Added Clone Support (NB);
51	* 23-Sep-2003 : Fixed Checkstyle issues (DG);
52	* 25-Sep-2003 : Oops, Range immutable, clone not necessary (NB);
53	* 05-May-2004 : Added constrain() and intersects() methods (DG);
54	* 18-May-2004 : Added expand() method (DG);
55	* ----- JFreeChart 1.0.x -----
56	* 11-Jan-2006 : Added new method expandToInclude(Range, double) (DG);
57	* 18-Dec-2007 : New methods intersects(Range) and scale(...) thanks to Sergei
58	*       Ivanov (DG);

```
59      * 08-Jan-2012 : New method combineIgnoringNaN() (DG);
60      * 23-Feb-2014 : Added isNaNRange() method (DG);
61      *
62      */
63
64      package org.jfree.data;
65
66      import java.io.Serializable;
67      import org.jfree.chart.util.ParamChecks;
68
69      /**
70       * Represents an immutable range of values.
71       */
72      public strictfp class Range implements Serializable {
73
74          /** For serialization. */
75          private static final long serialVersionUID = -906333695431863380L;
76
77          /** The lower bound of the range. */
78          private double lower;
```

79

80        `/** The upper bound of the range. */`

81        `private double upper;`

82

83        `/**`

84        `* Creates a new range.`

85        `*`

86        `* @param lower the lower bound (must be <= upper bound).`

87        `* @param upper the upper bound (must be >= lower bound).`

88        `*/`

89        `public Range(double lower, double upper) {`

90        [19](#)        `if (lower > upper) {`

91        [9](#)        `String msg = "Range(double, double): require lower (" + lower`

92        [11](#)        `+ ") <= upper (" + upper + ").";`

93        [1](#)        `throw new IllegalArgumentException(msg);`

94        `}`

95        [6](#)        `this.lower = lower;`

96        [6](#)        `this.upper = upper;`

97        `}`

98

99		/**
100		* Returns the lower bound for the range.
101		*
102		* @return The lower bound.
103		*/
104		public double getLowerBound() {
105	<a href="#">Z</a>	return this.lower;
106		}
107		
108		/**
109		* Returns the upper bound for the range.
110		*
111		* @return The upper bound.
112		*/
113		public double getUpperBound() {
114	<a href="#">Z</a>	return this.upper;
115		}
116		
117		/**
118		* Returns the length of the range.

119	*
120	* @return The length.
121	*/
122	public double getLength() {
123	<a href="#">19</a> return this.upper - this.lower;
124	}
125	
126	/**
127	* Returns the central value for the range.
128	*
129	* @return The central value.
130	*/
131	public double getCentralValue() {
132	<a href="#">47</a> return this.lower / 2.0 + this.upper / 2.0;
133	}
134	
135	/**
136	* Returns <code>true</code> if the range contains the specified value and
137	* <code>false</code> otherwise.
138	*

139		* @param value the value to lookup.
140		*
141		* @return <code>true</code> if the range contains the specified value.
142		*/
143		public boolean contains(double value) {
144	<a href="#">53</a>	return (value >= this.lower && value <= this.upper);
145		}
146		
147		/**
148		* Returns <code>true</code> if the range intersects with the specified
149		* range, and <code>false</code> otherwise.
150		*
151		* @param b0 the lower bound (should be <= b1).
152		* @param b1 the upper bound (should be >= b0).
153		*
154		* @return A boolean.
155		*/
156		public boolean intersects(double b0, double b1) {
157	<a href="#">19</a>	if (b0 <= this.lower) {
158	<a href="#">34</a>	return (b1 > this.lower);



```

159     }
160     else {
161         53     return (b0 < this.upper && b1 >= b0);
162     }
163 }
164
165 /**
166  * Returns true if the range intersects with the specified
167  * range, and false otherwise.
168  *
169  * @param range  another range (null not permitted).
170  *
171  * @return A boolean.
172  *
173  * @since 1.0.9
174  */
175 public boolean intersects(Range range) {
176     6     return intersects(range.getLowerBound(), range.getUpperBound());
177 }
178

```

179		/**
180		* Returns the value within the range that is closest to the specified
181		* value.
182		*
183		* @param value the value.
184		*
185		* @return The constrained value.
186		*/
187		public double constrain(double value) {
188	<a href="#">5</a>	double result = value;
189	<a href="#">14</a>	if (!contains(value)) {
190	<a href="#">19</a>	if (value > this.upper) {
191	<a href="#">5</a>	result = this.upper;
192		}
193	<a href="#">19</a>	else if (value < this.lower) {
194	<a href="#">5</a>	result = this.lower;
195		}
196		}
197	<a href="#">7</a>	return result;
198		}

199

200

```
/**
```

201

```
* Creates a new range by combining two existing ranges.
```

202

```
* <P>
```

203

```
* Note that:
```

204

```
* <ul>
```

205

```
* <li>either range can be <code>null</code>, in which case the other
```

206

```
*   range is returned;</li>
```

207

```
* <li>if both ranges are <code>null</code> the return value is
```

208

```
*   <code>null</code>.</li>
```

209

```
* </ul>
```

210

```
*
```

211

```
* @param range1 the first range (<code>null</code> permitted).
```

212

```
* @param range2 the second range (<code>null</code> permitted).
```

213

```
*
```

214

```
* @return A new range (possibly <code>null</code>).
```

215

```
*/
```

216

```
public static Range combine(Range range1, Range range2) {
```

217 4

```
    if (range1 == null) {
```

218 2

```
        return range2;
```

219		}
220	<a href="#">4</a>	if (range2 == null) {
221	<a href="#">2</a>	return range1;
222		}
223	<a href="#">4</a>	double l = Math.min(range1.getLowerBound(), range2.getLowerBound());
224	<a href="#">4</a>	double u = Math.max(range1.getUpperBound(), range2.getUpperBound());
225	<a href="#">13</a>	return new Range(l, u);
226		}
227		
228		/**
229		* Returns a new range that spans both <code>range1</code> and
230		* <code>range2</code> . This method has a special handling to ignore
231		* Double.NaN values.
232		*
233		* @param range1 the first range ( <code>null</code> permitted).
234		* @param range2 the second range ( <code>null</code> permitted).
235		*
236		* @return A new range (possibly <code>null</code> ).
237		*
238		* @since 1.0.15

239		<code>*/</code>
240		<code>public static Range combineIgnoringNaN(Range range1, Range range2) {</code>
241	<a href="#">4</a>	<code>if (range1 == null) {</code>
242	<a href="#">13</a>	<code>if (range2 != null &amp;&amp; range2.isNaNRange()) {</code>
243	<a href="#">1</a>	<code>return null;</code>
244		<code>}</code>
245	<a href="#">2</a>	<code>return range2;</code>
246		<code>}</code>
247	<a href="#">4</a>	<code>if (range2 == null) {</code>
248	<a href="#">9</a>	<code>if (range1.isNaNRange()) {</code>
249	<a href="#">1</a>	<code>return null;</code>
250		<code>}</code>
251	<a href="#">2</a>	<code>return range1;</code>
252		<code>}</code>
253	<a href="#">4</a>	<code>double l = min(range1.getLowerBound(), range2.getLowerBound());</code>
254	<a href="#">4</a>	<code>double u = max(range1.getUpperBound(), range2.getUpperBound());</code>
255	<a href="#">28</a>	<code>if (Double.isNaN(l) &amp;&amp; Double.isNaN(u)) {</code>
256	<a href="#">1</a>	<code>return null;</code>
257		<code>}</code>
258	<a href="#">13</a>	<code>return new Range(l, u);</code>

259		}
260		
261		/**
262		* Returns the minimum value. If either value is NaN, the other value is
263		* returned. If both are NaN, NaN is returned.
264		*
265		* @param d1 value 1.
266		* @param d2 value 2.
267		*
268		* @return The minimum of the two values.
269		*/
270		private static double min(double d1, double d2) {
271	<a href="#">14</a>	if (Double.isNaN(d1)) {
272	<a href="#">7</a>	return d2;
273		}
274	<a href="#">14</a>	if (Double.isNaN(d2)) {
275	<a href="#">7</a>	return d1;
276		}
277	<a href="#">14</a>	return Math.min(d1, d2);
278		}

279	
280	private static double max(double d1, double d2) {
281	<a href="#">14</a> if (Double.isNaN(d1)) {
282	<a href="#">7</a> return d2;
283	}
284	<a href="#">14</a> if (Double.isNaN(d2)) {
285	<a href="#">7</a> return d1;
286	}
287	<a href="#">14</a> return Math.max(d1, d2);
288	}
289	
290	/**
291	* Returns a range that includes all the values in the specified
292	* <code>range</code> AND the specified <code>value</code> .
293	*
294	* @param range the range ( <code>null</code> permitted).
295	* @param value the value that must be included.
296	*
297	* @return A range.
298	*

299		* @since 1.0.1
300		*/
301		public static Range expandToInclude(Range range, double value) {
302	<u>4</u>	if (range == null) {
303	<u>13</u>	return new Range(value, value);
304		}
305	<u>15</u>	if (value < range.getLowerBound()) {
306	<u>9</u>	return new Range(value, range.getUpperBound());
307		}
308	<u>15</u>	else if (value > range.getUpperBound()) {
309	<u>9</u>	return new Range(range.getLowerBound(), value);
310		}
311		else {
312	<u>2</u>	return range;
313		}
314		}
315		
316		/**
317		* Creates a new range by adding margins to an existing range.
318		*



319		* @param range the range (<code>>null</code> not permitted).
320		* @param lowerMargin the lower margin (expressed as a percentage of the
321		* range length).
322		* @param upperMargin the upper margin (expressed as a percentage of the
323		* range length).
324		*
325		* @return The expanded range.
326		*/
327		public static Range expand(Range range,
328		double lowerMargin, double upperMargin) {
329	<a href="#">1</a>	ParamChecks.checkNotNull(range, "range");
330	<a href="#">1</a>	double length = range.getLength();
331	<a href="#">25</a>	double lower = range.getLowerBound() - length * lowerMargin;
332	<a href="#">25</a>	double upper = range.getUpperBound() + length * upperMargin;
333	<a href="#">19</a>	if (lower > upper) {
334	<a href="#">45</a>	lower = lower / 2.0 + upper / 2.0;
335	<a href="#">5</a>	upper = lower;
336		}
337	<a href="#">13</a>	return new Range(lower, upper);
338		}

339

340

/\*\*

341

\* Shifts the range by the specified amount.

342

\*

343

\* @param base the base range (<code>null</code> not permitted).

344

\* @param delta the shift amount.

345

\*

346

\* @return A new range.

347

\*/

348

public static Range shift(Range base, double delta) {

349 [14](#)

return shift(base, delta, false);

350

}

351

352

/\*\*

353

\* Shifts the range by the specified amount.

354

\*

355

\* @param base the base range (<code>null</code> not permitted).

356

\* @param delta the shift amount.

357

\* @param allowZeroCrossing a flag that determines whether or not the

358

\* bounds of the range are allowed to cross

359		* zero after adjustment.
360		*
361		* @return A new range.
362		*/
363		public static Range shift(Range base, double delta,
364		boolean allowZeroCrossing) {
365	<a href="#">1</a>	ParamChecks.nullNotPermitted(base, "base");
366	<a href="#">13</a>	if (allowZeroCrossing) {
367	<a href="#">16</a>	return new Range(base.getLowerBound() + delta,
368	<a href="#">13</a>	base.getUpperBound() + delta);
369		}
370		else {
371	<a href="#">6</a>	return new Range(shiftWithNoZeroCrossing(base.getLowerBound(),
372	<a href="#">8</a>	delta), shiftWithNoZeroCrossing(base.getUpperBound(),
373	<a href="#">5</a>	delta));
374		}
375		}
376		
377		/**
378		* Returns the given <code>value</code> adjusted by <code>delta</code> but

379		* with a check to prevent the result from crossing <code>&gt;0.0&lt;/code&gt;.</code>
380		*
381		* @param value the value.
382		* @param delta the adjustment.
383		*
384		* @return The adjusted value.
385		*/
386		private static double shiftWithNoZeroCrossing(double value, double delta) {
387	<a href="#">19</a>	if (value > 0.0) {
388	<a href="#">26</a>	return Math.max(value + delta, 0.0);
389		}
390	<a href="#">19</a>	else if (value < 0.0) {
391	<a href="#">26</a>	return Math.min(value + delta, 0.0);
392		}
393		else {
394	<a href="#">19</a>	return value + delta;
395		}
396		}
397		
398		/**

399		* Scales the range by the specified factor.
400		*
401		* @param base the base range (<code>null</code> not permitted).
402		* @param factor the scaling factor (must be non-negative).
403		*
404		* @return A new range.
405		*
406		* @since 1.0.9
407		*/
408		public static Range scale(Range base, double factor) {
409	<a href="#">1</a>	ParamChecks.nullNotPermitted(base, "base");
410	<a href="#">19</a>	if (factor < 0) {
411	<a href="#">1</a>	throw new IllegalArgumentException("Negative 'factor' argument.");
412		}
413	<a href="#">16</a>	return new Range(base.getLowerBound() * factor,
414	<a href="#">13</a>	base.getUpperBound() * factor);
415		}
416		
417		/**
418		* Tests this object for equality with an arbitrary object.

419		*
420		* @param obj the object to test against (<code>null</code> permitted).
421		*
422		* @return A boolean.
423		*/
424		@Override
425		public boolean equals(Object obj) {
426	<a href="#">8</a>	if (!(obj instanceof Range)) {
427	<a href="#">7</a>	return false;
428		}
429		Range range = (Range) obj;
430	<a href="#">18</a>	if (!(this.lower == range.lower)) {
431	<a href="#">7</a>	return false;
432		}
433	<a href="#">18</a>	if (!(this.upper == range.upper)) {
434	<a href="#">7</a>	return false;
435		}
436	<a href="#">8</a>	return true;
437		}
438		

```
439     /**
440      * Returns true if both the lower and upper bounds are
441      * Double.NaN, and false otherwise.
442      *
443      * @return A boolean.
444      *
445      * @since 1.0.18
446      */
447     public boolean isNaNRange() {
448         43 return Double.isNaN(this.lower) && Double.isNaN(this.upper);
449     }
450
451     /**
452      * Returns a hash code.
453      *
454      * @return A hash code.
455      */
456     @Override
457     public int hashCode() {
458         int result;
```

459		long temp;
460	<a href="#">6</a>	temp = Double.doubleToLongBits(this.lower);
461	<a href="#">19</a>	result = (int) (temp ^ (temp >>> 32));
462	<a href="#">6</a>	temp = Double.doubleToLongBits(this.upper);
463	<a href="#">45</a>	result = 29 * result + (int) (temp ^ (temp >>> 32));
464	<a href="#">7</a>	return result;
465		}
466		
467		/**
468		* Returns a string representation of this Range.
469		*
470		* @return A String "Range[lower,upper]" where lower=lower range and
471		* upper=upper range.
472		*/
473		@Override
474		public String toString() {
475	<a href="#">22</a>	return ("Range[" + this.lower + "," + this.upper + "]");
476		}
477		
478		}



## Mutations

90

1. changed conditional boundary → KILLED
2. negated conditional → KILLED
3. removed conditional - replaced comparison check with false → KILLED
4. removed conditional - replaced comparison check with true → KILLED
5. Negated double local variable number 1 → KILLED
6. Negated double local variable number 3 → KILLED
7. Less or equal to less than → KILLED
8. Less or equal to greater than → KILLED
9. Less or equal to greater or equal → KILLED
10. Less or equal to equal → KILLED
11. Less or equal to not equal → KILLED
12. Incremented (a++) double local variable number 1 → KILLED
13. Incremented (a++) double local variable number 3 → KILLED
14. Decrementd (a--) double local variable number 1 → KILLED
15. Decrementd (a--) double local variable number 3 → KILLED
16. Incremented (++a) double local variable number 1 → KILLED
17. Incremented (++a) double local variable number 3 → KILLED
18. Decrementd (--a) double local variable number 1 → KILLED
19. Decrementd (--a) double local variable number 3 → KILLED

91

1. removed call to java/lang/StringBuilder::<init> → KILLED
2. removed call to java/lang/StringBuilder::append → KILLED
3. removed call to java/lang/StringBuilder::toString → SURVIVED
4. replaced call to java/lang/StringBuilder::append with receiver → SURVIVED
5. Negated double local variable number 1 → SURVIVED
6. Incremented (a++) double local variable number 1 → SURVIVED
7. Decrementd (a--) double local variable number 1 → SURVIVED
8. Incremented (++a) double local variable number 1 → SURVIVED
9. Decrementd (--a) double local variable number 1 → SURVIVED

92

1. removed call to java/lang/StringBuilder::append → KILLED
2. removed call to java/lang/StringBuilder::append → KILLED
3. removed call to java/lang/StringBuilder::append → KILLED
4. replaced call to java/lang/StringBuilder::append with receiver → SURVIVED
5. replaced call to java/lang/StringBuilder::append with receiver → SURVIVED
6. replaced call to java/lang/StringBuilder::append with receiver → SURVIVED
7. Negated double local variable number 3 → SURVIVED
8. Incremented (a++) double local variable number 3 → SURVIVED
9. Decrementd (a--) double local variable number 3 → SURVIVED
10. Incremented (++a) double local variable number 3 → SURVIVED
11. Decrementd (--a) double local variable number 3 → SURVIVED

93

1. removed call to java/lang/IllegalArgumentException::<init> → KILLED

[95](#)

1. Removed assignment to member variable lower → KILLED
2. Negated double local variable number 1 → KILLED
3. Incremented (a++) double local variable number 1 → SURVIVED
4. Decrementd (a--) double local variable number 1 → SURVIVED
5. Incremented (++a) double local variable number 1 → KILLED
6. Decrementd (--a) double local variable number 1 → KILLED

[96](#)

1. Removed assignment to member variable upper → KILLED
2. Negated double local variable number 3 → KILLED
3. Incremented (a++) double local variable number 3 → SURVIVED
4. Decrementd (a--) double local variable number 3 → SURVIVED
5. Incremented (++a) double local variable number 3 → KILLED
6. Decrementd (--a) double local variable number 3 → KILLED

[105](#)

1. replaced double return with 0.0d for org/jfree/data/Range::getLowerBound → KILLED
2. replaced return of double value with -(x + 1) for org/jfree/data/Range::getLowerBound → KILLED
3. Negated double field lower → KILLED
4. Incremented (a++) double field lower → KILLED
5. Decrementd (a--) double field lower → KILLED
6. Incremented (++a) double field lower → KILLED
7. Decrementd (--a) double fieldlower → KILLED

[114](#)

1. replaced double return with 0.0d for org/jfree/data/Range::getUpperBound → KILLED
2. replaced return of double value with -(x + 1) for org/jfree/data/Range::getUpperBound → KILLED
3. Negated double field upper → KILLED
4. Incremented (a++) double field upper → SURVIVED
5. Decrementd (a--) double field upper → SURVIVED
6. Incremented (++a) double field upper → KILLED
7. Decrementd (--a) double fieldupper → KILLED

[123](#)

1. Replaced double subtraction with addition → KILLED
2. replaced double return with 0.0d for org/jfree/data/Range::getLength → KILLED
3. replaced return of double value with -(x + 1) for org/jfree/data/Range::getLength → KILLED
4. Negated double field upper → KILLED
5. Negated double field lower → KILLED
6. Replaced double operation with first member → KILLED
7. Replaced double operation by second member → KILLED
8. Replaced double subtraction with addition → KILLED
9. Replaced double subtraction with multiplication → KILLED
10. Replaced double subtraction with division → KILLED

11. Replaced double subtraction with modulus → KILLED
12. Incremented (a++) double field upper → KILLED
13. Incremented (a++) double field lower → KILLED
14. Decrementd (a--) double field upper → KILLED
15. Decrementd (a--) double field lower → KILLED
16. Incremented (++a) double field upper → KILLED
17. Incremented (++a) double field lower → KILLED
18. Decrementd (--a) double fieldupper → KILLED
19. Decrementd (--a) double fieldlower → KILLED

1. Substituted 2.0 with 1.0 → KILLED
2. Substituted 2.0 with 1.0 → KILLED
3. Replaced double division with multiplication → KILLED
4. Replaced double division with multiplication → KILLED
5. Replaced double addition with subtraction → KILLED
6. replaced double return with 0.0d for org/jfree/data/Range::getCentralValue → KILLED
7. replaced return of double value with -(x + 1) for org/jfree/data/Range::getCentralValue → KILLED
8. Negated double field lower → KILLED
9. Negated double field upper → KILLED
10. Replaced double operation with first member → KILLED
11. Replaced double operation with first member → KILLED
12. Replaced double operation with first member → KILLED
13. Replaced double operation by second member → KILLED
14. Replaced double operation by second member → KILLED
15. Replaced double operation by second member → KILLED
16. Replaced double division with multiplication → KILLED
17. Replaced double division with multiplication → KILLED
18. Replaced double addition with subtraction → KILLED
19. Replaced double division with modulus → KILLED
20. Replaced double division with modulus → KILLED
21. Replaced double addition with multiplication → KILLED
22. Replaced double division with addition → KILLED
23. Replaced double division with addition → KILLED
24. Replaced double addition with division → KILLED
25. Replaced double division with subtraction → KILLED
26. Replaced double division with subtraction → KILLED
27. Replaced double addition with modulus → KILLED
28. Substituted 2.0 with 1.0 → KILLED
29. Substituted 2.0 with 1.0 → KILLED
30. Substituted 2.0 with 0.0 → KILLED
31. Substituted 2.0 with 0.0 → KILLED
32. Substituted 2.0 with -1.0 → KILLED
33. Substituted 2.0 with -1.0 → KILLED

- 34. Substituted 2.0 with -2.0 → KILLED
- 35. Substituted 2.0 with -2.0 → KILLED
- 36. Substituted 2.0 with 3.0 → KILLED
- 37. Substituted 2.0 with 3.0 → KILLED
- 38. Substituted 2.0 with 1.0 → KILLED
- 39. Substituted 2.0 with 1.0 → KILLED
- 40. Incremented (a++) double field lower → SURVIVED
- 41. Incremented (a++) double field upper → SURVIVED
- 42. Decrementd (a--) double field lower → SURVIVED
- 43. Decrementd (a--) double field upper → SURVIVED
- 44. Incremented (++a) double field lower → KILLED
- 45. Incremented (++a) double field upper → KILLED
- 46. Decrementd (--a) double fieldlower → KILLED
- 47. Decrementd (--a) double fieldupper → KILLED
- 1. replaced boolean return with false for org/jfree/data/Range::contains → KILLED
- 2. replaced boolean return with true for org/jfree/data/Range::contains → KILLED
- 3. changed conditional boundary → SURVIVED
- 4. changed conditional boundary → SURVIVED
- 5. Substituted 1 with 0 → KILLED
- 6. Substituted 0 with 1 → KILLED
- 7. negated conditional → KILLED
- 8. negated conditional → KILLED
- 9. removed conditional - replaced comparison check with false → KILLED
- 10. removed conditional - replaced comparison check with false → KILLED
- 11. removed conditional - replaced comparison check with true → KILLED
- 12. removed conditional - replaced comparison check with true → KILLED
- 13. replaced return of integer sized value with (x == 0 ? 1 : 0) → KILLED
- 14. replaced return of integer sized value with (x == 0 ? 1 : 0) → KILLED
- 15. Negated double local variable number 1 → KILLED
- 16. Negated double field lower → KILLED
- 17. Negated double local variable number 1 → KILLED
- 18. Negated double field upper → KILLED
- 19. Substituted 0 with 1 → KILLED
- 20. Substituted 1 with 0 → KILLED
- 21. Substituted 1 with -1 → SURVIVED
- 22. Substituted 0 with -1 → KILLED
- 23. Substituted 1 with -1 → SURVIVED
- 24. Substituted 1 with 2 → KILLED
- 25. Substituted 0 with 1 → KILLED
- 26. Substituted 1 with 0 → KILLED
- 27. Substituted 0 with -1 → KILLED
- 28. Less than to less or equal → SURVIVED

29. greater than to less than → KILLED  
 30. Less than to greater than → KILLED  
 31. greater than to less or equal → KILLED  
 32. Less than to greater or equal → KILLED  
 33. greater than to greater or equal → SURVIVED  
 34. Less than to equal → KILLED  
 35. greater than to equal → KILLED  
 36. Less than to not equal → KILLED  
 37. greater than to not equal → KILLED  
 38. Incremented (a++) double local variable number 1 → KILLED  
 39. Incremented (a++) double field lower → KILLED  
 40. Incremented (a++) double local variable number 1 → SURVIVED  
 41. Incremented (a++) double field upper → KILLED  
 42. Decrementd (a--) double local variable number 1 → KILLED  
 43. Decrementd (a--) double field lower → KILLED  
 44. Decrementd (a--) double local variable number 1 → SURVIVED  
 45. Decrementd (a--) double field upper → KILLED  
 46. Incremented (++a) double local variable number 1 → KILLED  
 47. Incremented (++a) double field lower → KILLED  
 48. Incremented (++a) double local variable number 1 → KILLED  
 49. Incremented (++a) double field upper → KILLED  
 50. Decrementd (--a) double local variable number 1 → KILLED  
 51. Decrementd (--a) double fieldlower → KILLED  
 52. Decrementd (--a) double local variable number 1 → KILLED  
 53. Decrementd (--a) double fieldupper → KILLED

1. changed conditional boundary → SURVIVED  
 2. negated conditional → KILLED  
 3. removed conditional - replaced comparison check with false → KILLED  
 4. removed conditional - replaced comparison check with true → SURVIVED  
 5. Negated double local variable number 1 → SURVIVED  
 6. Negated double field lower → KILLED  
 7. greater than to less than → KILLED  
 8. greater than to less or equal → KILLED  
 9. greater than to greater or equal → SURVIVED  
 10. greater than to equal → SURVIVED  
 11. greater than to not equal → KILLED  
 12. Incremented (a++) double local variable number 1 → KILLED  
 13. Incremented (a++) double field lower → KILLED  
 14. Decrementd (a--) double local variable number 1 → KILLED  
 15. Decrementd (a--) double field lower → KILLED  
 16. Incremented (++a) double local variable number 1 → KILLED  
 17. Incremented (++a) double field lower → KILLED  
 18. Decrementd (--a) double local variable number 1 → SURVIVED  
 19. Decrementd (--a) double fieldlower → KILLED

[158](#)

1. replaced boolean return with false for org/jfree/data/Range::intersects → KILLED
2. replaced boolean return with true for org/jfree/data/Range::intersects → KILLED
3. changed conditional boundary → KILLED
4. Substituted 1 with 0 → KILLED
5. Substituted 0 with 1 → KILLED
6. negated conditional → KILLED
7. removed conditional - replaced comparison check with false → KILLED
8. removed conditional - replaced comparison check with true → KILLED
9. replaced return of integer sized value with (x == 0 ? 1 : 0) → KILLED
10. replaced return of integer sized value with (x == 0 ? 1 : 0) → KILLED
11. Negated double local variable number 3 → KILLED
12. Negated double field lower → KILLED
13. Substituted 0 with 1 → KILLED
14. Substituted 1 with 0 → KILLED
15. Substituted 1 with -1 → SURVIVED
16. Substituted 0 with -1 → KILLED
17. Substituted 1 with -1 → SURVIVED
18. Substituted 1 with 2 → KILLED
19. Substituted 0 with 1 → KILLED
20. Substituted 1 with 0 → KILLED
21. Substituted 0 with -1 → KILLED
22. Less or equal to less than → KILLED
23. Less or equal to greater than → KILLED
24. Less or equal to greater or equal → KILLED
25. Less or equal to equal → SURVIVED
26. Less or equal to not equal → KILLED
27. Incremented (a++) double local variable number 3 → SURVIVED
28. Incremented (a++) double field lower → SURVIVED
29. Decrementd (a--) double local variable number 3 → SURVIVED
30. Decrementd (a--) double field lower → SURVIVED
31. Incremented (++a) double local variable number 3 → KILLED
32. Incremented (++a) double field lower → KILLED
33. Decrementd (--a) double local variable number 3 → KILLED
34. Decrementd (--a) double fieldlower → KILLED

[161](#)

1. replaced boolean return with false for org/jfree/data/Range::intersects → KILLED
2. replaced boolean return with true for org/jfree/data/Range::intersects → KILLED
3. changed conditional boundary → SURVIVED
4. changed conditional boundary → SURVIVED
5. Substituted 1 with 0 → KILLED
6. Substituted 0 with 1 → KILLED



7. negated conditional → KILLED  
8. negated conditional → KILLED  
9. removed conditional - replaced comparison check with false → KILLED  
10. removed conditional - replaced comparison check with false → KILLED  
11. removed conditional - replaced comparison check with true → SURVIVED  
12. removed conditional - replaced comparison check with true → KILLED  
13. replaced return of integer sized value with (x == 0 ? 1 : 0) → KILLED  
14. replaced return of integer sized value with (x == 0 ? 1 : 0) → KILLED  
15. Negated double local variable number 1 → SURVIVED  
16. Negated double field upper → KILLED  
17. Negated double local variable number 3 → KILLED  
18. Negated double local variable number 1 → KILLED  
19. Substituted 0 with 1 → KILLED  
20. Substituted 1 with 0 → KILLED  
21. Substituted 1 with -1 → SURVIVED  
22. Substituted 0 with -1 → KILLED  
23. Substituted 1 with -1 → SURVIVED  
24. Substituted 1 with 2 → KILLED  
25. Substituted 0 with 1 → KILLED  
26. Substituted 1 with 0 → KILLED  
27. Substituted 0 with -1 → KILLED  
28. greater or equal to less than → KILLED  
29. Less than to less or equal → SURVIVED  
30. greater or equal to less or equal → KILLED  
31. Less than to greater than → KILLED  
32. greater or equal to greater than → SURVIVED  
33. Less than to greater or equal → KILLED  
34. greater or equal to equal → SURVIVED  
35. Less than to equal → KILLED  
36. greater or equal to not equal → KILLED  
37. Less than to not equal → KILLED  
38. Incremented (a++) double local variable number 1 → SURVIVED  
39. Incremented (a++) double field upper → SURVIVED  
40. Incremented (a++) double local variable number 3 → SURVIVED  
41. Incremented (a++) double local variable number 1 → SURVIVED  
42. Decrementd (a--) double local variable number 1 → KILLED  
43. Decrementd (a--) double field upper → SURVIVED  
44. Decrementd (a--) double local variable number 3 → SURVIVED  
45. Decrementd (a--) double local variable number 1 → SURVIVED  
46. Incremented (++a) double local variable number 1 → KILLED  
47. Incremented (++a) double field upper → SURVIVED  
48. Incremented (++a) double local variable number 3 → KILLED  
49. Incremented (++a) double local variable number 1 → SURVIVED  
50. Decrementd (--a) double local variable number 1 → KILLED  
51. Decrementd (--a) double fieldupper → KILLED

	52. Decrementd (--a) double local variable number 3 → SURVIVED
	53. Decrementd (--a) double local variable number 1 → KILLED
<a href="#"><u>176</u></a>	1. replaced boolean return with false for org/jfree/data/Range::intersects → NO_COVERAGE 2. replaced boolean return with true for org/jfree/data/Range::intersects → NO_COVERAGE 3. removed call to org/jfree/data/Range::getLowerBound → NO_COVERAGE 4. removed call to org/jfree/data/Range::getUpperBound → NO_COVERAGE 5. removed call to org/jfree/data/Range::intersects → NO_COVERAGE 6. replaced return of integer sized value with (x == 0 ? 1 : 0) → NO_COVERAGE
<a href="#"><u>188</u></a>	1. Negated double local variable number 1 → KILLED 2. Incremented (a++) double local variable number 1 → SURVIVED 3. Decrementd (a--) double local variable number 1 → SURVIVED 4. Incremented (++a) double local variable number 1 → KILLED 5. Decrementd (--a) double local variable number 1 → KILLED
<a href="#"><u>189</u></a>	1. negated conditional → KILLED 2. removed call to org/jfree/data/Range::contains → SURVIVED 3. removed conditional - replaced equality check with false → KILLED 4. removed conditional - replaced equality check with true → SURVIVED 5. Negated double local variable number 1 → SURVIVED 6. not equal to less than → SURVIVED 7. not equal to less or equal → KILLED 8. not equal to greater than → SURVIVED 9. not equal to greater or equal → KILLED 10. not equal to equal → KILLED 11. Incremented (a++) double local variable number 1 → SURVIVED 12. Decrementd (a--) double local variable number 1 → SURVIVED 13. Incremented (++a) double local variable number 1 → SURVIVED 14. Decrementd (--a) double local variable number 1 → SURVIVED
<a href="#"><u>190</u></a>	1. changed conditional boundary → SURVIVED 2. negated conditional → KILLED 3. removed conditional - replaced comparison check with false → KILLED 4. removed conditional - replaced comparison check with true → KILLED 5. Negated double local variable number 1 → KILLED 6. Negated double field upper → SURVIVED 7. Less or equal to less than → SURVIVED 8. Less or equal to greater than → KILLED 9. Less or equal to greater or equal → KILLED 10. Less or equal to equal → KILLED 11. Less or equal to not equal → KILLED 12. Incremented (a++) double local variable number 1 → SURVIVED 13. Incremented (a++) double field upper → KILLED



[191](#)

14. Decrementd (a--) double local variable number 1 → SURVIVED
15. Decrementd (a--) double field upper → KILLED
16. Incremented (++a) double local variable number 1 → SURVIVED
17. Incremented (++a) double field upper → KILLED
18. Decrementd (--a) double local variable number 1 → SURVIVED
19. Decrementd (--a) double fieldupper → KILLED

1. Negated double field upper → KILLED
2. Incremented (a++) double field upper → SURVIVED
3. Decrementd (a--) double field upper → SURVIVED
4. Incremented (++a) double field upper → KILLED
5. Decrementd (--a) double fieldupper → KILLED

[193](#)

1. changed conditional boundary → SURVIVED
2. negated conditional → KILLED
3. removed conditional - replaced comparison check with false → KILLED
4. removed conditional - replaced comparison check with true → SURVIVED
5. Negated double local variable number 1 → KILLED
6. Negated double field lower → SURVIVED
7. greater or equal to less than → KILLED
8. greater or equal to less or equal → KILLED
9. greater or equal to greater than → SURVIVED
10. greater or equal to equal → SURVIVED
11. greater or equal to not equal → KILLED
12. Incremented (a++) double local variable number 1 → SURVIVED
13. Incremented (a++) double field lower → KILLED
14. Decrementd (a--) double local variable number 1 → SURVIVED
15. Decrementd (a--) double field lower → KILLED
16. Incremented (++a) double local variable number 1 → SURVIVED
17. Incremented (++a) double field lower → KILLED
18. Decrementd (--a) double local variable number 1 → SURVIVED
19. Decrementd (--a) double fieldlower → KILLED

[194](#)

1. Negated double field lower → SURVIVED
2. Incremented (a++) double field lower → SURVIVED
3. Decrementd (a--) double field lower → SURVIVED
4. Incremented (++a) double field lower → KILLED
5. Decrementd (--a) double fieldlower → KILLED

[197](#)

1. replaced double return with 0.0d for org/jfree/data/Range::constrain → KILLED
2. replaced return of double value with -(x + 1) for org/jfree/data/Range::constrain → KILLED
3. Negated double local variable number 3 → KILLED
4. Incremented (a++) double local variable number 3 → SURVIVED
5. Decrementd (a--) double local variable number 3 → SURVIVED

	6. Incremented (++a) double local variable number 3 → KILLED
	7. Decrementd (--a) double local variable number 3 → KILLED
<a href="#"><u>217</u></a>	1. negated conditional → KILLED
	2. removed conditional - replaced equality check with false → SURVIVED
	3. removed conditional - replaced equality check with true → KILLED
	4. not equal to equal → KILLED
<a href="#"><u>218</u></a>	1. replaced return value with null for org/jfree/data/Range::combine → SURVIVED
	2. mutated return of Object value for org/jfree/data/Range::combine to ( if (x != null) null else throw new RuntimeException ) → KILLED
<a href="#"><u>220</u></a>	1. negated conditional → KILLED
	2. removed conditional - replaced equality check with false → KILLED
	3. removed conditional - replaced equality check with true → KILLED
	4. not equal to equal → KILLED
<a href="#"><u>221</u></a>	1. replaced return value with null for org/jfree/data/Range::combine → KILLED
	2. mutated return of Object value for org/jfree/data/Range::combine to ( if (x != null) null else throw new RuntimeException ) → KILLED
<a href="#"><u>223</u></a>	1. replaced call to java/lang/Math::min with argument → KILLED
	2. removed call to org/jfree/data/Range::getLowerBound → KILLED
	3. removed call to org/jfree/data/Range::getLowerBound → SURVIVED
	4. removed call to java/lang/Math::min → KILLED
<a href="#"><u>224</u></a>	1. replaced call to java/lang/Math::max with argument → KILLED
	2. removed call to org/jfree/data/Range::getUpperBound → KILLED
	3. removed call to org/jfree/data/Range::getUpperBound → KILLED
	4. removed call to java/lang/Math::max → KILLED
<a href="#"><u>225</u></a>	1. removed call to org/jfree/data/Range::<init> → KILLED
	2. replaced return value with null for org/jfree/data/Range::combine → KILLED
	3. mutated return of Object value for org/jfree/data/Range::combine to ( if (x != null) null else throw new RuntimeException ) → KILLED
	4. Negated double local variable number 2 → KILLED
	5. Negated double local variable number 4 → KILLED
	6. Incremented (a++) double local variable number 2 → SURVIVED
	7. Incremented (a++) double local variable number 4 → SURVIVED
	8. Decrementd (a--) double local variable number 2 → SURVIVED
	9. Decrementd (a--) double local variable number 4 → SURVIVED
	10. Incremented (++a) double local variable number 2 → KILLED
	11. Incremented (++a) double local variable number 4 → KILLED
	12. Decrementd (--a) double local variable number 2 → KILLED

	13. Decrementd (--a) double local variable number 4 → KILLED
<a href="#"><u>241</u></a>	<ul style="list-style-type: none"> <li>1. negated conditional → KILLED</li> <li>2. removed conditional - replaced equality check with false → KILLED</li> <li>3. removed conditional - replaced equality check with true → KILLED</li> <li>4. not equal to equal → KILLED</li> </ul>
<a href="#"><u>242</u></a>	<ul style="list-style-type: none"> <li>1. negated conditional → KILLED</li> <li>2. negated conditional → KILLED</li> <li>3. removed call to org/jfree/data/Range::isNaNRange → KILLED</li> <li>4. removed conditional - replaced equality check with false → KILLED</li> <li>5. removed conditional - replaced equality check with false → KILLED</li> <li>6. removed conditional - replaced equality check with true → KILLED</li> <li>7. removed conditional - replaced equality check with true → SURVIVED</li> <li>8. equal to less than → SURVIVED</li> <li>9. equal to less or equal → SURVIVED</li> <li>10. equal to greater than → KILLED</li> <li>11. equal to greater or equal → KILLED</li> <li>12. equal to not equal → KILLED</li> <li>13. equal to not equal → KILLED</li> </ul>
<a href="#"><u>243</u></a>	1. mutated return of Object value for org/jfree/data/Range::combineIgnoringNaN to ( if (x != null) null else throw new RuntimeException ) → KILLED
<a href="#"><u>245</u></a>	<ul style="list-style-type: none"> <li>1. replaced return value with null for org/jfree/data/Range::combineIgnoringNaN → SURVIVED</li> <li>2. mutated return of Object value for org/jfree/data/Range::combineIgnoringNaN to ( if (x != null) null else throw new RuntimeException ) → KILLED</li> </ul>
<a href="#"><u>247</u></a>	<ul style="list-style-type: none"> <li>1. negated conditional → KILLED</li> <li>2. removed conditional - replaced equality check with false → KILLED</li> <li>3. removed conditional - replaced equality check with true → KILLED</li> <li>4. not equal to equal → KILLED</li> </ul>
<a href="#"><u>248</u></a>	<ul style="list-style-type: none"> <li>1. negated conditional → KILLED</li> <li>2. removed call to org/jfree/data/Range::isNaNRange → KILLED</li> <li>3. removed conditional - replaced equality check with false → KILLED</li> <li>4. removed conditional - replaced equality check with true → KILLED</li> <li>5. equal to less than → KILLED</li> <li>6. equal to less or equal → SURVIVED</li> <li>7. equal to greater than → KILLED</li> <li>8. equal to greater or equal → KILLED</li> <li>9. equal to not equal → KILLED</li> </ul>

[249](#)

1. mutated return of Object value for  
org/jfree/data/Range::combineIgnoringNaN to ( if (x != null) null else throw  
new RuntimeException ) → KILLED

[251](#)

1. replaced return value with null for  
org/jfree/data/Range::combineIgnoringNaN → KILLED  
2. mutated return of Object value for  
org/jfree/data/Range::combineIgnoringNaN to ( if (x != null) null else throw  
new RuntimeException ) → KILLED

[253](#)

1. replaced call to org/jfree/data/Range::min with argument → KILLED  
2. removed call to org/jfree/data/Range::getLowerBound → KILLED  
3. removed call to org/jfree/data/Range::getLowerBound → KILLED  
4. removed call to org/jfree/data/Range::min → KILLED

[254](#)

1. replaced call to org/jfree/data/Range::max with argument → KILLED  
2. removed call to org/jfree/data/Range::getUpperBound → KILLED  
3. removed call to org/jfree/data/Range::getUpperBound → KILLED  
4. removed call to org/jfree/data/Range::max → KILLED

[255](#)

1. negated conditional → KILLED  
2. negated conditional → KILLED  
3. removed call to java/lang/Double::isNaN → KILLED  
4. removed call to java/lang/Double::isNaN → KILLED  
5. removed conditional - replaced equality check with false → KILLED  
6. removed conditional - replaced equality check with false → KILLED  
7. removed conditional - replaced equality check with true → SURVIVED  
8. removed conditional - replaced equality check with true → SURVIVED  
9. Negated double local variable number 2 → SURVIVED  
10. Negated double local variable number 4 → SURVIVED  
11. equal to less than → SURVIVED  
12. equal to less than → SURVIVED  
13. equal to less or equal → SURVIVED  
14. equal to less or equal → SURVIVED  
15. equal to greater than → KILLED  
16. equal to greater than → KILLED  
17. equal to greater or equal → KILLED  
18. equal to greater or equal → KILLED  
19. equal to not equal → KILLED  
20. equal to not equal → KILLED  
21. Incremented (a++) double local variable number 2 → KILLED  
22. Incremented (a++) double local variable number 4 → SURVIVED  
23. Decrementd (a--) double local variable number 2 → KILLED  
24. Decrementd (a--) double local variable number 4 → SURVIVED  
25. Incremented (++a) double local variable number 2 → KILLED  
26. Incremented (++a) double local variable number 4 → SURVIVED

[256](#)

27. Decrementd (--a) double local variable number 2 → KILLED

28. Decrementd (--a) double local variable number 4 → SURVIVED

1. mutated return of Object value for  
org/jfree/data/Range::combineIgnoringNaN to ( if (x != null) null else throw  
new RuntimeException ) → KILLED

[258](#)

1. removed call to org/jfree/data/Range::<init> → KILLED

2. replaced return value with null for

org/jfree/data/Range::combineIgnoringNaN → KILLED

3. mutated return of Object value for

org/jfree/data/Range::combineIgnoringNaN to ( if (x != null) null else throw  
new RuntimeException ) → KILLED

4. Negated double local variable number 2 → KILLED

5. Negated double local variable number 4 → KILLED

6. Incremented (a++) double local variable number 2 → SURVIVED

7. Incremented (a++) double local variable number 4 → SURVIVED

8. Decrementd (a--) double local variable number 2 → SURVIVED

9. Decrementd (a--) double local variable number 4 → SURVIVED

10. Incremented (++a) double local variable number 2 → KILLED

11. Incremented (++a) double local variable number 4 → KILLED

12. Decrementd (--a) double local variable number 2 → KILLED

13. Decrementd (--a) double local variable number 4 → KILLED

[271](#)

1. negated conditional → KILLED

2. removed call to java/lang/Double::isNaN → SURVIVED

3. removed conditional - replaced equality check with false → SURVIVED

4. removed conditional - replaced equality check with true → KILLED

5. Negated double local variable number 0 → SURVIVED

6. equal to less than → KILLED

7. equal to less or equal → SURVIVED

8. equal to greater than → KILLED

9. equal to greater or equal → SURVIVED

10. equal to not equal → KILLED

11. Incremented (a++) double local variable number 0 → KILLED

12. Decrementd (a--) double local variable number 0 → KILLED

13. Incremented (++a) double local variable number 0 → KILLED

14. Decrementd (--a) double local variable number 0 → KILLED

[272](#)

1. replaced double return with 0.0d for org/jfree/data/Range::min → KILLED

2. replaced return of double value with -(x + 1) for org/jfree/data/Range::min  
→ KILLED

3. Negated double local variable number 2 → SURVIVED

4. Incremented (a++) double local variable number 2 → SURVIVED

5. Decrementd (a--) double local variable number 2 → SURVIVED

6. Incremented (++a) double local variable number 2 → SURVIVED

	7. Decrementd (--a) double local variable number 2 → SURVIVED
	1. negated conditional → KILLED
	2. removed call to java/lang/Double::isNaN → KILLED
	3. removed conditional - replaced equality check with false → KILLED
	4. removed conditional - replaced equality check with true → KILLED
	5. Negated double local variable number 2 → SURVIVED
	6. equal to less than → KILLED
<a href="#">274</a>	7. equal to less or equal → SURVIVED
	8. equal to greater than → KILLED
	9. equal to greater or equal → KILLED
	10. equal to not equal → KILLED
	11. Incremented (a++) double local variable number 2 → KILLED
	12. Decrementd (a--) double local variable number 2 → KILLED
	13. Incremented (++a) double local variable number 2 → KILLED
	14. Decrementd (--a) double local variable number 2 → KILLED
	1. replaced double return with 0.0d for org/jfree/data/Range::min → KILLED
	2. replaced return of double value with -(x + 1) for org/jfree/data/Range::min → KILLED
<a href="#">275</a>	3. Negated double local variable number 0 → KILLED
	4. Incremented (a++) double local variable number 0 → SURVIVED
	5. Decrementd (a--) double local variable number 0 → SURVIVED
	6. Incremented (++a) double local variable number 0 → KILLED
	7. Decrementd (--a) double local variable number 0 → KILLED
	1. replaced call to java/lang/Math::min with argument → SURVIVED
	2. removed call to java/lang/Math::min → KILLED
	3. replaced double return with 0.0d for org/jfree/data/Range::min → KILLED
	4. replaced return of double value with -(x + 1) for org/jfree/data/Range::min → KILLED
	5. Negated double local variable number 0 → SURVIVED
	6. Negated double local variable number 2 → KILLED
<a href="#">277</a>	7. Incremented (a++) double local variable number 0 → SURVIVED
	8. Incremented (a++) double local variable number 2 → SURVIVED
	9. Decrementd (a--) double local variable number 0 → SURVIVED
	10. Decrementd (a--) double local variable number 2 → SURVIVED
	11. Incremented (++a) double local variable number 0 → SURVIVED
	12. Incremented (++a) double local variable number 2 → KILLED
	13. Decrementd (--a) double local variable number 0 → SURVIVED
	14. Decrementd (--a) double local variable number 2 → KILLED
	1. negated conditional → KILLED
<a href="#">281</a>	2. removed call to java/lang/Double::isNaN → SURVIVED
	3. removed conditional - replaced equality check with false → SURVIVED
	4. removed conditional - replaced equality check with true → KILLED



[282](#)

- 5. Negated double local variable number 0 → SURVIVED
- 6. equal to less than → KILLED
- 7. equal to less or equal → SURVIVED
- 8. equal to greater than → KILLED
- 9. equal to greater or equal → SURVIVED
- 10. equal to not equal → KILLED
- 11. Incremented (a++) double local variable number 0 → KILLED
- 12. Decrementd (a--) double local variable number 0 → KILLED
- 13. Incremented (++a) double local variable number 0 → KILLED
- 14. Decrementd (--a) double local variable number 0 → KILLED

- 1. replaced double return with 0.0d for org/jfree/data/Range::max → KILLED
- 2. replaced return of double value with -(x + 1) for org/jfree/data/Range::max → KILLED

- 3. Negated double local variable number 2 → SURVIVED
- 4. Incremented (a++) double local variable number 2 → SURVIVED
- 5. Decrementd (a--) double local variable number 2 → SURVIVED
- 6. Incremented (++a) double local variable number 2 → SURVIVED
- 7. Decrementd (--a) double local variable number 2 → SURVIVED

[284](#)

- 1. negated conditional → KILLED
- 2. removed call to java/lang/Double::isNaN → SURVIVED
- 3. removed conditional - replaced equality check with false → SURVIVED
- 4. removed conditional - replaced equality check with true → KILLED
- 5. Negated double local variable number 2 → SURVIVED
- 6. equal to less than → KILLED
- 7. equal to less or equal → SURVIVED
- 8. equal to greater than → KILLED
- 9. equal to greater or equal → SURVIVED
- 10. equal to not equal → KILLED
- 11. Incremented (a++) double local variable number 2 → KILLED
- 12. Decrementd (a--) double local variable number 2 → KILLED
- 13. Incremented (++a) double local variable number 2 → KILLED
- 14. Decrementd (--a) double local variable number 2 → KILLED

[285](#)

- 1. replaced double return with 0.0d for org/jfree/data/Range::max → NO\_COVERAGE
- 2. replaced return of double value with -(x + 1) for org/jfree/data/Range::max → NO\_COVERAGE
- 3. Negated double local variable number 0 → NO\_COVERAGE
- 4. Incremented (a++) double local variable number 0 → NO\_COVERAGE
- 5. Decrementd (a--) double local variable number 0 → NO\_COVERAGE
- 6. Incremented (++a) double local variable number 0 → NO\_COVERAGE
- 7. Decrementd (--a) double local variable number 0 → NO\_COVERAGE

[287](#)

1. replaced call to java/lang/Math::max with argument → KILLED
2. removed call to java/lang/Math::max → KILLED
3. replaced double return with 0.0d for org/jfree/data/Range::max → KILLED
4. replaced return of double value with -(x + 1) for org/jfree/data/Range::max → KILLED
5. Negated double local variable number 0 → KILLED
6. Negated double local variable number 2 → KILLED
7. Incremented (a++) double local variable number 0 → SURVIVED
8. Incremented (a++) double local variable number 2 → SURVIVED
9. Decrementd (a--) double local variable number 0 → SURVIVED
10. Decrementd (a--) double local variable number 2 → SURVIVED
11. Incremented (++a) double local variable number 0 → KILLED
12. Incremented (++a) double local variable number 2 → KILLED
13. Decrementd (--a) double local variable number 0 → KILLED
14. Decrementd (--a) double local variable number 2 → KILLED

[302](#)

1. negated conditional → KILLED
2. removed conditional - replaced equality check with false → KILLED
3. removed conditional - replaced equality check with true → KILLED
4. not equal to equal → KILLED

[303](#)

1. removed call to org/jfree/data/Range::<init> → KILLED
2. replaced return value with null for org/jfree/data/Range::expandToInclude → KILLED
3. mutated return of Object value for org/jfree/data/Range::expandToInclude to ( if (x != null) null else throw new RuntimeException ) → KILLED
4. Negated double local variable number 1 → SURVIVED
5. Negated double local variable number 1 → SURVIVED
6. Incremented (a++) double local variable number 1 → KILLED
7. Incremented (a++) double local variable number 1 → SURVIVED
8. Decrementd (a--) double local variable number 1 → KILLED
9. Decrementd (a--) double local variable number 1 → SURVIVED
10. Incremented (++a) double local variable number 1 → KILLED
11. Incremented (++a) double local variable number 1 → KILLED
12. Decrementd (--a) double local variable number 1 → KILLED
13. Decrementd (--a) double local variable number 1 → KILLED

[305](#)

1. changed conditional boundary → SURVIVED
2. negated conditional → KILLED
3. removed call to org/jfree/data/Range::getLowerBound → SURVIVED
4. removed conditional - replaced comparison check with false → KILLED
5. removed conditional - replaced comparison check with true → KILLED
6. Negated double local variable number 1 → KILLED
7. greater or equal to less than → KILLED
8. greater or equal to less or equal → KILLED
9. greater or equal to greater than → SURVIVED



306

10. greater or equal to equal → KILLED
11. greater or equal to not equal → KILLED
12. Incremented (a++) double local variable number 1 → KILLED
13. Decrementd (a--) double local variable number 1 → KILLED
14. Incremented (++a) double local variable number 1 → KILLED
15. Decrementd (--a) double local variable number 1 → KILLED

1. removed call to org/jfree/data/Range::<init> → KILLED
2. removed call to org/jfree/data/Range::getUpperBound → KILLED
3. replaced return value with null for org/jfree/data/Range::expandToInclude → KILLED
4. mutated return of Object value for org/jfree/data/Range::expandToInclude to ( if (x != null) null else throw new RuntimeException ) → KILLED
5. Negated double local variable number 1 → KILLED
6. Incremented (a++) double local variable number 1 → SURVIVED
7. Decrementd (a--) double local variable number 1 → SURVIVED
8. Incremented (++a) double local variable number 1 → KILLED
9. Decrementd (--a) double local variable number 1 → KILLED

308

1. changed conditional boundary → SURVIVED
2. negated conditional → KILLED
3. removed call to org/jfree/data/Range::getUpperBound → SURVIVED
4. removed conditional - replaced comparison check with false → KILLED
5. removed conditional - replaced comparison check with true → KILLED
6. Negated double local variable number 1 → KILLED
7. Less or equal to less than → SURVIVED
8. Less or equal to greater than → KILLED
9. Less or equal to greater or equal → KILLED
10. Less or equal to equal → KILLED
11. Less or equal to not equal → KILLED
12. Incremented (a++) double local variable number 1 → KILLED
13. Decrementd (a--) double local variable number 1 → KILLED
14. Incremented (++a) double local variable number 1 → KILLED
15. Decrementd (--a) double local variable number 1 → KILLED

309

1. removed call to org/jfree/data/Range::<init> → KILLED
2. removed call to org/jfree/data/Range::getLowerBound → KILLED
3. replaced return value with null for org/jfree/data/Range::expandToInclude → KILLED
4. mutated return of Object value for org/jfree/data/Range::expandToInclude to ( if (x != null) null else throw new RuntimeException ) → KILLED
5. Negated double local variable number 1 → KILLED
6. Incremented (a++) double local variable number 1 → SURVIVED
7. Decrementd (a--) double local variable number 1 → SURVIVED
8. Incremented (++a) double local variable number 1 → KILLED
9. Decrementd (--a) double local variable number 1 → KILLED

[312](#)

1. replaced return value with null for org/jfree/data/Range::expandToInclude → KILLED
2. mutated return of Object value for org/jfree/data/Range::expandToInclude to ( if (x != null) null else throw new RuntimeException ) → KILLED

[329](#)

1. removed call to org/jfree/chart/util/ParamChecks::nullNotPermitted → SURVIVED

[330](#)

1. removed call to org/jfree/data/Range::getLength → KILLED

[331](#)

1. Replaced double multiplication with division → KILLED
2. Replaced double subtraction with addition → KILLED
3. removed call to org/jfree/data/Range::getLowerBound → SURVIVED
4. Negated double local variable number 5 → KILLED
5. Negated double local variable number 1 → KILLED
6. Replaced double operation with first member → KILLED
7. Replaced double operation with first member → KILLED
8. Replaced double operation by second member → SURVIVED
9. Replaced double operation by second member → KILLED
10. Replaced double multiplication with division → KILLED
11. Replaced double subtraction with addition → KILLED
12. Replaced double multiplication with modulus → KILLED
13. Replaced double subtraction with multiplication → KILLED
14. Replaced double multiplication with addition → KILLED
15. Replaced double subtraction with division → KILLED
16. Replaced double multiplication with subtraction → KILLED
17. Replaced double subtraction with modulus → KILLED
18. Incremented (a++) double local variable number 5 → KILLED
19. Incremented (a++) double local variable number 1 → SURVIVED
20. Decrementd (a--) double local variable number 5 → KILLED
21. Decrementd (a--) double local variable number 1 → SURVIVED
22. Incremented (++a) double local variable number 5 → KILLED
23. Incremented (++a) double local variable number 1 → KILLED
24. Decrementd (--a) double local variable number 5 → KILLED
25. Decrementd (--a) double local variable number 1 → KILLED

[332](#)

1. Replaced double multiplication with division → KILLED
2. Replaced double addition with subtraction → KILLED
3. removed call to org/jfree/data/Range::getUpperBound → KILLED
4. Negated double local variable number 5 → KILLED
5. Negated double local variable number 3 → KILLED
6. Replaced double operation with first member → KILLED
7. Replaced double operation with first member → KILLED
8. Replaced double operation by second member → SURVIVED
9. Replaced double operation by second member → KILLED

10. Replaced double multiplication with division → KILLED  
11. Replaced double addition with subtraction → KILLED  
12. Replaced double multiplication with modulus → KILLED  
13. Replaced double addition with multiplication → KILLED  
14. Replaced double multiplication with addition → KILLED  
15. Replaced double addition with division → KILLED  
16. Replaced double multiplication with subtraction → KILLED  
17. Replaced double addition with modulus → KILLED  
18. Incremented (a++) double local variable number 5 → SURVIVED  
19. Incremented (a++) double local variable number 3 → SURVIVED  
20. Decrementd (a--) double local variable number 5 → SURVIVED  
21. Decrementd (a--) double local variable number 3 → SURVIVED  
22. Incremented (++a) double local variable number 5 → KILLED  
23. Incremented (++a) double local variable number 3 → KILLED  
24. Decrementd (--a) double local variable number 5 → KILLED  
25. Decrementd (--a) double local variable number 3 → KILLED

1. changed conditional boundary → SURVIVED  
2. negated conditional → KILLED  
3. removed conditional - replaced comparison check with false → KILLED  
4. removed conditional - replaced comparison check with true → KILLED  
5. Negated double local variable number 7 → KILLED  
6. Negated double local variable number 9 → KILLED  
7. Less or equal to less than → SURVIVED  
8. Less or equal to greater than → KILLED  
9. Less or equal to greater or equal → KILLED  
10. Less or equal to equal → KILLED  
11. Less or equal to not equal → KILLED  
12. Incremented (a++) double local variable number 7 → KILLED  
13. Incremented (a++) double local variable number 9 → KILLED  
14. Decrementd (a--) double local variable number 7 → KILLED  
15. Decrementd (a--) double local variable number 9 → KILLED  
16. Incremented (++a) double local variable number 7 → KILLED  
17. Incremented (++a) double local variable number 9 → KILLED  
18. Decrementd (--a) double local variable number 7 → KILLED  
19. Decrementd (--a) double local variable number 9 → KILLED

1. Substituted 2.0 with 1.0 → KILLED  
2. Substituted 2.0 with 1.0 → KILLED  
3. Replaced double division with multiplication → KILLED  
4. Replaced double division with multiplication → KILLED  
5. Replaced double addition with subtraction → KILLED  
6. Negated double local variable number 7 → KILLED  
7. Negated double local variable number 9 → KILLED  
8. Replaced double operation with first member → KILLED

[333](#)

[334](#)

9. Replaced double operation with first member → KILLED  
 10. Replaced double operation with first member → KILLED  
 11. Replaced double operation by second member → KILLED  
 12. Replaced double operation by second member → KILLED  
 13. Replaced double operation by second member → KILLED  
 14. Replaced double division with multiplication → KILLED  
 15. Replaced double division with multiplication → KILLED  
 16. Replaced double addition with subtraction → KILLED  
 17. Replaced double division with modulus → KILLED  
 18. Replaced double division with modulus → KILLED  
 19. Replaced double addition with multiplication → KILLED  
 20. Replaced double division with addition → KILLED  
 21. Replaced double division with addition → KILLED  
 22. Replaced double addition with division → KILLED  
 23. Replaced double division with subtraction → KILLED  
 24. Replaced double division with subtraction → KILLED  
 25. Replaced double addition with modulus → KILLED  
 26. Substituted 2.0 with 1.0 → KILLED  
 27. Substituted 2.0 with 1.0 → KILLED  
 28. Substituted 2.0 with 0.0 → KILLED  
 29. Substituted 2.0 with 0.0 → KILLED  
 30. Substituted 2.0 with -1.0 → KILLED  
 31. Substituted 2.0 with -1.0 → KILLED  
 32. Substituted 2.0 with -2.0 → KILLED  
 33. Substituted 2.0 with -2.0 → KILLED  
 34. Substituted 2.0 with 3.0 → KILLED  
 35. Substituted 2.0 with 3.0 → KILLED  
 36. Substituted 2.0 with 1.0 → KILLED  
 37. Substituted 2.0 with 1.0 → KILLED  
 38. Incremented (a++) double local variable number 7 → SURVIVED  
 39. Incremented (a++) double local variable number 9 → SURVIVED  
 40. Decrementd (a--) double local variable number 7 → SURVIVED  
 41. Decrementd (a--) double local variable number 9 → SURVIVED  
 42. Incremented (++a) double local variable number 7 → KILLED  
 43. Incremented (++a) double local variable number 9 → KILLED  
 44. Decrementd (--a) double local variable number 7 → KILLED  
 45. Decrementd (--a) double local variable number 9 → KILLED

[335](#)

1. Negated double local variable number 7 → KILLED  
 2. Incremented (a++) double local variable number 7 → KILLED  
 3. Decrementd (a--) double local variable number 7 → KILLED  
 4. Incremented (++a) double local variable number 7 → KILLED  
 5. Decrementd (--a) double local variable number 7 → KILLED

[337](#)

1. removed call to org/jfree/data/Range::<init> → KILLED

[349](#)

2. replaced return value with null for org/jfree/data/Range::expand → KILLED  
3. mutated return of Object value for org/jfree/data/Range::expand to ( if (x != null) null else throw new RuntimeException ) → KILLED  
4. Negated double local variable number 7 → KILLED  
5. Negated double local variable number 9 → KILLED  
6. Incremented (a++) double local variable number 7 → SURVIVED  
7. Incremented (a++) double local variable number 9 → SURVIVED  
8. Decrementd (a--) double local variable number 7 → SURVIVED  
9. Decrementd (a--) double local variable number 9 → SURVIVED  
10. Incremented (++a) double local variable number 7 → KILLED  
11. Incremented (++a) double local variable number 9 → KILLED  
12. Decrementd (--a) double local variable number 7 → KILLED  
13. Decrementd (--a) double local variable number 9 → KILLED

1. replaced call to org/jfree/data/Range::shift with argument → KILLED  
2. Substituted 0 with 1 → SURVIVED  
3. removed call to org/jfree/data/Range::shift → KILLED  
4. replaced return value with null for org/jfree/data/Range::shift → KILLED  
5. mutated return of Object value for org/jfree/data/Range::shift to ( if (x != null) null else throw new RuntimeException ) → KILLED  
6. Negated double local variable number 1 → KILLED  
7. Substituted 0 with 1 → SURVIVED  
8. Substituted 0 with -1 → SURVIVED  
9. Substituted 0 with 1 → SURVIVED  
10. Substituted 0 with -1 → SURVIVED  
11. Incremented (a++) double local variable number 1 → SURVIVED  
12. Decrementd (a--) double local variable number 1 → SURVIVED  
13. Incremented (++a) double local variable number 1 → KILLED  
14. Decrementd (--a) double local variable number 1 → KILLED

[365](#)

1. removed call to org/jfree/chart/util/ParamChecks::nullNotPermitted → SURVIVED

[366](#)

1. negated conditional → SURVIVED  
2. removed conditional - replaced equality check with false → SURVIVED  
3. removed conditional - replaced equality check with true → SURVIVED  
4. Negated integer local variable number 3 → SURVIVED  
5. equal to less than → SURVIVED  
6. equal to less or equal → SURVIVED  
7. equal to greater than → SURVIVED  
8. equal to greater or equal → SURVIVED  
9. equal to not equal → SURVIVED  
10. Incremented (a++) integer local variable number 3 → SURVIVED  
11. Decrementd (a--) integer local variable number 3 → SURVIVED  
12. Incremented (++a) integer local variable number 3 → SURVIVED  
13. Decrementd (--a) integer local variable number 3 → SURVIVED

[367](#)

1. removed call to org/jfree/data/Range::<init> → NO\_COVERAGE
2. Replaced double addition with subtraction → NO\_COVERAGE
3. removed call to org/jfree/data/Range::getLowerBound → NO\_COVERAGE
4. replaced return value with null for org/jfree/data/Range::shift → NO\_COVERAGE
5. mutated return of Object value for org/jfree/data/Range::shift to ( if (x != null) null else throw new RuntimeException ) → NO\_COVERAGE
6. Negated double local variable number 1 → NO\_COVERAGE
7. Replaced double operation with first member → NO\_COVERAGE
8. Replaced double operation by second member → NO\_COVERAGE
9. Replaced double addition with subtraction → NO\_COVERAGE
10. Replaced double addition with multiplication → NO\_COVERAGE
11. Replaced double addition with division → NO\_COVERAGE
12. Replaced double addition with modulus → NO\_COVERAGE
13. Incremented (a++) double local variable number 1 → NO\_COVERAGE
14. Decrementd (a--) double local variable number 1 → NO\_COVERAGE
15. Incremented (++a) double local variable number 1 → NO\_COVERAGE
16. Decrementd (--a) double local variable number 1 → NO\_COVERAGE

[368](#)

1. Replaced double addition with subtraction → NO\_COVERAGE
2. removed call to org/jfree/data/Range::getUpperBound → NO\_COVERAGE
3. Negated double local variable number 1 → NO\_COVERAGE
4. Replaced double operation with first member → NO\_COVERAGE
5. Replaced double operation by second member → NO\_COVERAGE
6. Replaced double addition with subtraction → NO\_COVERAGE
7. Replaced double addition with multiplication → NO\_COVERAGE
8. Replaced double addition with division → NO\_COVERAGE
9. Replaced double addition with modulus → NO\_COVERAGE
10. Incremented (a++) double local variable number 1 → NO\_COVERAGE
11. Decrementd (a--) double local variable number 1 → NO\_COVERAGE
12. Incremented (++a) double local variable number 1 → NO\_COVERAGE
13. Decrementd (--a) double local variable number 1 → NO\_COVERAGE

[371](#)

1. replaced call to org/jfree/data/Range::shiftWithNoZeroCrossing with argument → SURVIVED
2. removed call to org/jfree/data/Range::<init> → KILLED
3. removed call to org/jfree/data/Range::getLowerBound → SURVIVED
4. removed call to org/jfree/data/Range::shiftWithNoZeroCrossing → KILLED
5. replaced return value with null for org/jfree/data/Range::shift → KILLED
6. mutated return of Object value for org/jfree/data/Range::shift to ( if (x != null) null else throw new RuntimeException ) → KILLED

[372](#)

1. replaced call to org/jfree/data/Range::shiftWithNoZeroCrossing with argument → KILLED
2. removed call to org/jfree/data/Range::getUpperBound → KILLED
3. removed call to org/jfree/data/Range::shiftWithNoZeroCrossing → KILLED



[373](#)

4. Negated double local variable number 1 → KILLED
5. Incremented (a++) double local variable number 1 → KILLED
6. Decrementd (a--) double local variable number 1 → KILLED
7. Incremented (++a) double local variable number 1 → KILLED
8. Decrementd (--a) double local variable number 1 → KILLED

1. Negated double local variable number 1 → KILLED
2. Incremented (a++) double local variable number 1 → SURVIVED
3. Decrementd (a--) double local variable number 1 → SURVIVED
4. Incremented (++a) double local variable number 1 → KILLED
5. Decrementd (--a) double local variable number 1 → KILLED

[387](#)

1. changed conditional boundary → SURVIVED
2. Substituted 0.0 with 1.0 → SURVIVED
3. negated conditional → SURVIVED
4. removed conditional - replaced comparison check with false → SURVIVED
5. removed conditional - replaced comparison check with true → SURVIVED
6. Negated double local variable number 0 → SURVIVED
7. Substituted 0.0 with 1.0 → SURVIVED
8. Substituted 0.0 with -1.0 → SURVIVED
9. Substituted 0.0 with 1.0 → SURVIVED
10. Substituted 0.0 with -1.0 → SURVIVED
11. Less or equal to less than → SURVIVED
12. Less or equal to greater than → SURVIVED
13. Less or equal to greater or equal → SURVIVED
14. Less or equal to equal → SURVIVED
15. Less or equal to not equal → SURVIVED
16. Incremented (a++) double local variable number 0 → KILLED
17. Decrementd (a--) double local variable number 0 → KILLED
18. Incremented (++a) double local variable number 0 → KILLED
19. Decrementd (--a) double local variable number 0 → KILLED

[388](#)

1. replaced call to java/lang/Math::max with argument → KILLED
2. Substituted 0.0 with 1.0 → SURVIVED
3. Replaced double addition with subtraction → KILLED
4. removed call to java/lang/Math::max → KILLED
5. replaced double return with 0.0d for  
org/jfree/data/Range::shiftWithNoZeroCrossing → KILLED
6. replaced return of double value with -(x + 1) for  
org/jfree/data/Range::shiftWithNoZeroCrossing → KILLED
7. Negated double local variable number 0 → KILLED
8. Negated double local variable number 2 → KILLED
9. Replaced double operation with first member → KILLED
10. Replaced double operation by second member → KILLED
11. Replaced double addition with subtraction → KILLED
12. Replaced double addition with multiplication → KILLED

[390](#)

13. Replaced double addition with division → KILLED
14. Replaced double addition with modulus → KILLED
15. Substituted 0.0 with 1.0 → SURVIVED
16. Substituted 0.0 with -1.0 → SURVIVED
17. Substituted 0.0 with 1.0 → SURVIVED
18. Substituted 0.0 with -1.0 → SURVIVED
19. Incremented (a++) double local variable number 0 → SURVIVED
20. Incremented (a++) double local variable number 2 → SURVIVED
21. Decrementd (a--) double local variable number 0 → SURVIVED
22. Decrementd (a--) double local variable number 2 → SURVIVED
23. Incremented (++a) double local variable number 0 → KILLED
24. Incremented (++a) double local variable number 2 → KILLED
25. Decrementd (--a) double local variable number 0 → KILLED
26. Decrementd (--a) double local variable number 2 → KILLED

1. changed conditional boundary → KILLED
2. Substituted 0.0 with 1.0 → KILLED
3. negated conditional → KILLED
4. removed conditional - replaced comparison check with false → SURVIVED
5. removed conditional - replaced comparison check with true → KILLED
6. Negated double local variable number 0 → SURVIVED
7. Substituted 0.0 with 1.0 → KILLED
8. Substituted 0.0 with -1.0 → SURVIVED
9. Substituted 0.0 with 1.0 → KILLED
10. Substituted 0.0 with -1.0 → SURVIVED
11. greater or equal to less than → KILLED
12. greater or equal to less or equal → SURVIVED
13. greater or equal to greater than → KILLED
14. greater or equal to equal → SURVIVED
15. greater or equal to not equal → KILLED
16. Incremented (a++) double local variable number 0 → KILLED
17. Decrementd (a--) double local variable number 0 → KILLED
18. Incremented (++a) double local variable number 0 → KILLED
19. Decrementd (--a) double local variable number 0 → KILLED

[391](#)

1. replaced call to java/lang/Math::min with argument → NO\_COVERAGE
2. Substituted 0.0 with 1.0 → NO\_COVERAGE
3. Replaced double addition with subtraction → NO\_COVERAGE
4. removed call to java/lang/Math::min → NO\_COVERAGE
5. replaced double return with 0.0d for  
org/jfree/data/Range::shiftWithNoZeroCrossing → NO\_COVERAGE
6. replaced return of double value with -(x + 1) for  
org/jfree/data/Range::shiftWithNoZeroCrossing → NO\_COVERAGE
7. Negated double local variable number 0 → NO\_COVERAGE
8. Negated double local variable number 2 → NO\_COVERAGE



[394](#)

9. Replaced double operation with first member → NO\_COVERAGE  
10. Replaced double operation by second member → NO\_COVERAGE  
11. Replaced double addition with subtraction → NO\_COVERAGE  
12. Replaced double addition with multiplication → NO\_COVERAGE  
13. Replaced double addition with division → NO\_COVERAGE  
14. Replaced double addition with modulus → NO\_COVERAGE  
15. Substituted 0.0 with 1.0 → NO\_COVERAGE  
16. Substituted 0.0 with -1.0 → NO\_COVERAGE  
17. Substituted 0.0 with 1.0 → NO\_COVERAGE  
18. Substituted 0.0 with -1.0 → NO\_COVERAGE  
19. Incremented (a++) double local variable number 0 → NO\_COVERAGE  
20. Incremented (a++) double local variable number 2 → NO\_COVERAGE  
21. Decrementd (a--) double local variable number 0 → NO\_COVERAGE  
22. Decrementd (a--) double local variable number 2 → NO\_COVERAGE  
23. Incremented (++a) double local variable number 0 → NO\_COVERAGE  
24. Incremented (++a) double local variable number 2 → NO\_COVERAGE  
25. Decrementd (--a) double local variable number 0 → NO\_COVERAGE  
26. Decrementd (--a) double local variable number 2 → NO\_COVERAGE

1. Replaced double addition with subtraction → KILLED  
2. replaced double return with 0.0d for  
org/jfree/data/Range::shiftWithNoZeroCrossing → KILLED  
3. replaced return of double value with -(x + 1) for  
org/jfree/data/Range::shiftWithNoZeroCrossing → KILLED  
4. Negated double local variable number 0 → SURVIVED  
5. Negated double local variable number 2 → KILLED  
6. Replaced double operation with first member → KILLED  
7. Replaced double operation by second member → SURVIVED  
8. Replaced double addition with subtraction → KILLED  
9. Replaced double addition with multiplication → KILLED  
10. Replaced double addition with division → KILLED  
11. Replaced double addition with modulus → KILLED  
12. Incremented (a++) double local variable number 0 → SURVIVED  
13. Incremented (a++) double local variable number 2 → SURVIVED  
14. Decrementd (a--) double local variable number 0 → SURVIVED  
15. Decrementd (a--) double local variable number 2 → SURVIVED  
16. Incremented (++a) double local variable number 0 → KILLED  
17. Incremented (++a) double local variable number 2 → KILLED  
18. Decrementd (--a) double local variable number 0 → KILLED  
19. Decrementd (--a) double local variable number 2 → KILLED

[409](#)

1. removed call to org/jfree/chart/util/ParamChecks::nullNotPermitted → SURVIVED

[410](#)

1. changed conditional boundary → SURVIVED  
2. Substituted 0.0 with 1.0 → SURVIVED

3. negated conditional → SURVIVED
4. removed conditional - replaced comparison check with false → SURVIVED
5. removed conditional - replaced comparison check with true → SURVIVED
6. Negated double local variable number 1 → SURVIVED
7. Substituted 0.0 with 1.0 → SURVIVED
8. Substituted 0.0 with -1.0 → SURVIVED
9. Substituted 0.0 with 1.0 → SURVIVED
10. Substituted 0.0 with -1.0 → SURVIVED
11. greater or equal to less than → SURVIVED
12. greater or equal to less or equal → SURVIVED
13. greater or equal to greater than → SURVIVED
14. greater or equal to equal → SURVIVED
15. greater or equal to not equal → SURVIVED
16. Incremented (a++) double local variable number 1 → SURVIVED
17. Decrementd (a--) double local variable number 1 → SURVIVED
18. Incremented (++a) double local variable number 1 → SURVIVED
19. Decrementd (--a) double local variable number 1 → SURVIVED

[411](#)

1. removed call to java/lang/IllegalArgumentException::<init> → KILLED

1. removed call to org/jfree/data/Range::<init> → NO\_COVERAGE
2. Replaced double multiplication with division → NO\_COVERAGE
3. removed call to org/jfree/data/Range::getLowerBound → NO\_COVERAGE
4. replaced return value with null for org/jfree/data/Range::scale → NO\_COVERAGE
5. mutated return of Object value for org/jfree/data/Range::scale to ( if (x != null) null else throw new RuntimeException ) → NO\_COVERAGE
6. Negated double local variable number 1 → NO\_COVERAGE
7. Replaced double operation with first member → NO\_COVERAGE
8. Replaced double operation by second member → NO\_COVERAGE
9. Replaced double multiplication with division → NO\_COVERAGE
10. Replaced double multiplication with modulus → NO\_COVERAGE
11. Replaced double multiplication with addition → NO\_COVERAGE
12. Replaced double multiplication with subtraction → NO\_COVERAGE
13. Incremented (a++) double local variable number 1 → NO\_COVERAGE
14. Decrementd (a--) double local variable number 1 → NO\_COVERAGE
15. Incremented (++a) double local variable number 1 → NO\_COVERAGE
16. Decrementd (--a) double local variable number 1 → NO\_COVERAGE

[413](#)

1. Replaced double multiplication with division → NO\_COVERAGE
2. removed call to org/jfree/data/Range::getUpperBound → NO\_COVERAGE
3. Negated double local variable number 1 → NO\_COVERAGE
4. Replaced double operation with first member → NO\_COVERAGE
5. Replaced double operation by second member → NO\_COVERAGE
6. Replaced double multiplication with division → NO\_COVERAGE

[414](#)

[426](#)

7. Replaced double multiplication with modulus → NO\_COVERAGE  
8. Replaced double multiplication with addition → NO\_COVERAGE  
9. Replaced double multiplication with subtraction → NO\_COVERAGE  
10. Incremented (a++) double local variable number 1 → NO\_COVERAGE  
11. Decrementd (a--) double local variable number 1 → NO\_COVERAGE  
12. Incremented (++a) double local variable number 1 → NO\_COVERAGE  
13. Decrementd (--a) double local variable number 1 → NO\_COVERAGE

1. negated conditional → KILLED  
2. removed conditional - replaced equality check with false → SURVIVED  
3. removed conditional - replaced equality check with true → KILLED  
4. not equal to less than → KILLED  
5. not equal to less or equal → KILLED  
6. not equal to greater than → SURVIVED  
7. not equal to greater or equal → SURVIVED  
8. not equal to equal → KILLED

[427](#)

1. replaced boolean return with true for org/jfree/data/Range::equals → NO\_COVERAGE  
2. Substituted 0 with 1 → NO\_COVERAGE  
3. replaced return of integer sized value with (x == 0 ? 1 : 0) → NO\_COVERAGE  
4. Substituted 0 with 1 → NO\_COVERAGE  
5. Substituted 0 with -1 → NO\_COVERAGE  
6. Substituted 0 with 1 → NO\_COVERAGE  
7. Substituted 0 with -1 → NO\_COVERAGE

[430](#)

1. negated conditional → KILLED  
2. removed conditional - replaced equality check with false → SURVIVED  
3. removed conditional - replaced equality check with true → KILLED  
4. Negated double field lower → KILLED  
5. Negated double field lower → KILLED  
6. equal to less than → KILLED  
7. equal to less or equal → SURVIVED  
8. equal to greater than → KILLED  
9. equal to greater or equal → SURVIVED  
10. equal to not equal → KILLED  
11. Incremented (a++) double field lower → KILLED  
12. Incremented (a++) double field lower → SURVIVED  
13. Decrementd (a--) double field lower → KILLED  
14. Decrementd (a--) double field lower → SURVIVED  
15. Incremented (++a) double field lower → KILLED  
16. Incremented (++a) double field lower → KILLED  
17. Decrementd (--a) double fieldlower → KILLED  
18. Decrementd (--a) double fieldlower → KILLED

[431](#)

1. replaced boolean return with true for org/jfree/data/Range::equals → NO\_COVERAGE
2. Substituted 0 with 1 → NO\_COVERAGE
3. replaced return of integer sized value with (x == 0 ? 1 : 0) → NO\_COVERAGE
4. Substituted 0 with 1 → NO\_COVERAGE
5. Substituted 0 with -1 → NO\_COVERAGE
6. Substituted 0 with 1 → NO\_COVERAGE
7. Substituted 0 with -1 → NO\_COVERAGE

[433](#)

1. negated conditional → KILLED
2. removed conditional - replaced equality check with false → SURVIVED
3. removed conditional - replaced equality check with true → KILLED
4. Negated double field upper → KILLED
5. Negated double field upper → KILLED
6. equal to less than → KILLED
7. equal to less or equal → SURVIVED
8. equal to greater than → KILLED
9. equal to greater or equal → SURVIVED
10. equal to not equal → KILLED
11. Incremented (a++) double field upper → KILLED
12. Incremented (a++) double field upper → SURVIVED
13. Decrementd (a--) double field upper → KILLED
14. Decrementd (a--) double field upper → SURVIVED
15. Incremented (++a) double field upper → KILLED
16. Incremented (++a) double field upper → KILLED
17. Decrementd (--a) double fieldupper → KILLED
18. Decrementd (--a) double fieldupper → KILLED

[434](#)

1. replaced boolean return with true for org/jfree/data/Range::equals → NO\_COVERAGE
2. Substituted 0 with 1 → NO\_COVERAGE
3. replaced return of integer sized value with (x == 0 ? 1 : 0) → NO\_COVERAGE
4. Substituted 0 with 1 → NO\_COVERAGE
5. Substituted 0 with -1 → NO\_COVERAGE
6. Substituted 0 with 1 → NO\_COVERAGE
7. Substituted 0 with -1 → NO\_COVERAGE

[436](#)

1. replaced boolean return with false for org/jfree/data/Range::equals → KILLED
2. Substituted 1 with 0 → KILLED
3. replaced return of integer sized value with (x == 0 ? 1 : 0) → KILLED
4. Substituted 1 with 0 → KILLED
5. Substituted 1 with -1 → SURVIVED
6. Substituted 1 with -1 → SURVIVED
7. Substituted 1 with 2 → KILLED
8. Substituted 1 with 0 → KILLED

1. replaced boolean return with false for org/jfree/data/Range::isNaNRange → KILLED
2. replaced boolean return with true for org/jfree/data/Range::isNaNRange → KILLED
3. Substituted 1 with 0 → KILLED
4. Substituted 0 with 1 → KILLED
5. negated conditional → KILLED
6. negated conditional → KILLED
7. removed call to java/lang/Double::isNaN → KILLED
8. removed call to java/lang/Double::isNaN → KILLED
9. removed conditional - replaced equality check with false → KILLED
10. removed conditional - replaced equality check with false → KILLED
11. removed conditional - replaced equality check with true → SURVIVED
12. removed conditional - replaced equality check with true → SURVIVED
13. replaced return of integer sized value with (x == 0 ? 1 : 0) → KILLED
14. replaced return of integer sized value with (x == 0 ? 1 : 0) → KILLED
15. Negated double field lower → SURVIVED
16. Negated double field upper → SURVIVED
17. Substituted 0 with 1 → KILLED
18. Substituted 1 with 0 → KILLED
19. Substituted 1 with -1 → SURVIVED
20. Substituted 0 with -1 → KILLED
21. Substituted 1 with -1 → SURVIVED
22. Substituted 1 with 2 → KILLED
23. Substituted 0 with 1 → KILLED
24. Substituted 1 with 0 → KILLED
25. Substituted 0 with -1 → KILLED
26. equal to less than → SURVIVED
27. equal to less than → SURVIVED
28. equal to less or equal → SURVIVED
29. equal to less or equal → SURVIVED
30. equal to greater than → KILLED
31. equal to greater than → KILLED
32. equal to greater or equal → KILLED
33. equal to greater or equal → KILLED
34. equal to not equal → KILLED
35. equal to not equal → KILLED
36. Incremented (a++) double field lower → SURVIVED
37. Incremented (a++) double field upper → SURVIVED
38. Decrementd (a--) double field lower → SURVIVED
39. Decrementd (a--) double field upper → SURVIVED
40. Incremented (++a) double field lower → SURVIVED
41. Incremented (++a) double field upper → SURVIVED
42. Decrementd (--a) double fieldlower → SURVIVED
43. Decrementd (--a) double fieldupper → SURVIVED

[460](#)

1. removed call to java/lang/Double::doubleToLongBits → SURVIVED
2. Negated double field lower → KILLED
3. Incremented (a++) double field lower → SURVIVED
4. Decrementd (a--) double field lower → SURVIVED
5. Incremented (++a) double field lower → KILLED
6. Decrementd (--a) double fieldlower → KILLED

[461](#)

1. Substituted 32 with 33 → SURVIVED
2. Replaced Unsigned Shift Right with Shift Left → SURVIVED
3. Replaced XOR with AND → SURVIVED
4. Negated long local variable number 1 → SURVIVED
5. Negated long local variable number 1 → SURVIVED
6. Substituted 32 with 1 → SURVIVED
7. Substituted 32 with 0 → SURVIVED
8. Substituted 32 with -1 → SURVIVED
9. Substituted 32 with -32 → SURVIVED
10. Substituted 32 with 33 → SURVIVED
11. Substituted 32 with 31 → SURVIVED
12. Incremented (a++) long local variable number 1 → SURVIVED
13. Incremented (a++) long local variable number 1 → SURVIVED
14. Decrementd (a--) long local variable number 1 → KILLED
15. Decrementd (a--) long local variable number 1 → SURVIVED
16. Incremented (++a) long local variable number 1 → KILLED
17. Incremented (++a) long local variable number 1 → SURVIVED
18. Decrementd (--a) long local variable number 2 → SURVIVED
19. Decrementd (--a) long local variable number 2 → KILLED

[462](#)

1. removed call to java/lang/Double::doubleToLongBits → KILLED
2. Negated double field upper → KILLED
3. Incremented (a++) double field upper → SURVIVED
4. Decrementd (a--) double field upper → SURVIVED
5. Incremented (++a) double field upper → KILLED
6. Decrementd (--a) double fieldupper → KILLED

[463](#)

1. Substituted 29 with 30 → SURVIVED
2. Substituted 32 with 33 → KILLED
3. Replaced integer multiplication with division → KILLED
4. Replaced Unsigned Shift Right with Shift Left → KILLED
5. Replaced XOR with AND → KILLED
6. Replaced integer addition with subtraction → KILLED
7. Negated integer local variable number 3 → SURVIVED
8. Negated long local variable number 1 → SURVIVED
9. Negated long local variable number 1 → KILLED
10. Replaced integer operation with first member → KILLED
11. Replaced integer operation with first member → KILLED
12. Replaced integer operation by second member → SURVIVED



[464](#)

13. Replaced integer operation by second member → SURVIVED
14. Replaced integer multiplication with division → KILLED
15. Replaced integer addition with subtraction → KILLED
16. Replaced integer multiplication with modulus → KILLED
17. Replaced integer addition with multiplication → KILLED
18. Replaced integer multiplication with addition → KILLED
19. Replaced integer addition with division → KILLED
20. Replaced integer multiplication with subtraction → KILLED
21. Replaced integer addition with modulus → KILLED
22. Substituted 29 with 1 → SURVIVED
23. Substituted 32 with 1 → KILLED
24. Substituted 29 with 0 → SURVIVED
25. Substituted 32 with 0 → KILLED
26. Substituted 29 with -1 → SURVIVED
27. Substituted 32 with -1 → KILLED
28. Substituted 29 with -29 → SURVIVED
29. Substituted 32 with -32 → SURVIVED
30. Substituted 29 with 30 → SURVIVED
31. Substituted 32 with 33 → KILLED
32. Substituted 29 with 28 → SURVIVED
33. Substituted 32 with 31 → KILLED
34. Incremented (a++) integer local variable number 3 → SURVIVED
35. Incremented (a++) long local variable number 1 → SURVIVED
36. Incremented (a++) long local variable number 1 → SURVIVED
37. Decrementd (a--) integer local variable number 3 → SURVIVED
38. Decrementd (a--) long local variable number 1 → KILLED
39. Decrementd (a--) long local variable number 1 → SURVIVED
40. Incremented (++a) integer local variable number 3 → KILLED
41. Incremented (++a) long local variable number 1 → KILLED
42. Incremented (++a) long local variable number 1 → SURVIVED
43. Decrementd (--a) integer local variable number 1 → KILLED
44. Decrementd (--a) long local variable number 2 → KILLED
45. Decrementd (--a) long local variable number 2 → KILLED

[475](#)

1. replaced int return with 0 for org/jfree/data/Range::hashCode → KILLED
  2. replaced return of integer sized value with (x == 0 ? 1 : 0) → KILLED
  3. Negated integer local variable number 3 → KILLED
  4. Incremented (a++) integer local variable number 3 → SURVIVED
  5. Decrementd (a--) integer local variable number 3 → SURVIVED
  6. Incremented (++a) integer local variable number 3 → KILLED
  7. Decrementd (--a) integer local variable number 1 → KILLED
- 
1. removed call to java/lang/StringBuilder::<init> → KILLED
  2. replaced return value with "" for org/jfree/data/Range::toString → KILLED
  3. removed call to java/lang/StringBuilder::append → KILLED

4. removed call to java/lang/StringBuilder::append → KILLED  
5. removed call to java/lang/StringBuilder::append → KILLED  
6. removed call to java/lang/StringBuilder::append → KILLED  
7. removed call to java/lang/StringBuilder::toString → KILLED  
8. mutated return of Object value for org/jfree/data/Range::toString to ( if (x != null) null else throw new RuntimeException ) → KILLED  
9. replaced call to java/lang/StringBuilder::append with receiver → KILLED  
10. replaced call to java/lang/StringBuilder::append with receiver → KILLED  
11. replaced call to java/lang/StringBuilder::append with receiver → KILLED  
12. replaced call to java/lang/StringBuilder::append with receiver → KILLED  
13. Negated double field lower → KILLED  
14. Negated double field upper → KILLED  
15. Incremented (a++) double field lower → SURVIVED  
16. Incremented (a++) double field upper → SURVIVED  
17. Decrementd (a--) double field lower → SURVIVED  
18. Decrementd (a--) double field upper → SURVIVED  
19. Incremented (++a) double field lower → KILLED  
20. Incremented (++a) double field upper → KILLED  
21. Decrementd (--a) double fieldlower → KILLED  
22. Decrementd (--a) double fieldupper → KILLED

## Active mutators

- ABS\_MUTATOR
- AOD\_1\_MUTATOR
- AOD\_2\_MUTATOR
- AOR\_1\_MUTATOR
- AOR\_2\_MUTATOR
- AOR\_3\_MUTATOR
- AOR\_4\_MUTATOR
- ARGUMENT\_PROPAGATION\_MUTATOR
- BOOLEAN\_FALSE\_RETURN
- BOOLEAN\_TRUE\_RETURN
- CONDITIONALS\_BOUNDARY\_MUTATOR
- CONSTRUCTOR\_CALL\_MUTATOR
- CRCR\_1\_MUTATOR
- CRCR\_2\_MUTATOR
- CRCR\_3\_MUTATOR
- CRCR\_4\_MUTATOR
- CRCR\_5\_MUTATOR
- CRCR\_6\_MUTATOR
- EMPTY\_RETURN\_VALUES
- EXPERIMENTAL\_BIGINTEGER\_MUTATOR
- EXPERIMENTAL\_MEMBER\_VARIABLE\_MUTATOR



- EXPERIMENTAL\_REMOVE\_SWITCH\_MUTATOR\_[0-99]
- EXPERIMENTAL\_SWITCH\_MUTATOR
- INCREMENTS\_MUTATOR
- INLINE\_CONSTANT\_MUTATOR
- INVERT\_NEGS\_MUTATOR
- MATH\_MUTATOR
- NAKED\_RECEIVER
- NEGATE\_CONDITIONALS\_MUTATOR
- NON\_VOID\_METHOD\_CALL\_MUTATOR
- NULL\_RETURN\_VALUES
- OBBN\_1\_MUTATOR
- OBBN\_2\_MUTATOR
- OBBN\_3\_MUTATOR
- PRIMITIVE\_RETURN\_VALS\_MUTATOR
- REMOVE\_CONDITIONALS\_EQUAL\_ELSE\_MUTATOR
- REMOVE\_CONDITIONALS\_EQUAL\_IF\_MUTATOR
- REMOVE\_CONDITIONALS\_ORDER\_ELSE\_MUTATOR
- REMOVE\_CONDITIONALS\_ORDER\_IF\_MUTATOR
- REMOVE\_INCREMENTS\_MUTATOR
- RETURN\_VALS\_MUTATOR
- ROR\_1\_MUTATOR
- ROR\_2\_MUTATOR
- ROR\_3\_MUTATOR
- ROR\_4\_MUTATOR
- ROR\_5\_MUTATOR
- UOI\_1\_MUTATOR
- UOI\_2\_MUTATOR
- UOI\_3\_MUTATOR
- UOI\_4\_MUTATOR
- VOID\_METHOD\_CALL\_MUTATOR

## Tests examined

- org.jfree.data.RangeTest (37 ms)

Report generated by [PIT](#) 1.4.11